Generative Models for Fast Calorimeter Simulation

LHCb case

Viktoria Chekalina[1,2,*], *Elena* Orlova[3,**], *Fedor* Ratnikov[1,2], *Dmitry* Ulyanov[3], *Andrey* Ustyuzhanin[1,2], and *Egor* Zakharov[3,***]

[1]NRU Higher School of Economics, Moscow, Russia
[2]Yandex School of Data Analysis, Moscow, Russia
[3]Skolkovo Institute of Science and Technology, Moscow, Russia

Abstract. Simulation is a key component of high energy physics. Historically simulation in particle physics relies on the Monte Carlo methods which require a tremendous amount of computation. These methods may have difficulties with the expected High Luminosity Large Hadron Collider (HL LHC) need, so the experiment is in urgent need of new fast simulation techniques. We introduce a new Deep Learning framework based on Generative Adversarial Networks which can be faster than traditional simulation methods by 5 order of magnitude with reasonable simulation accuracy. This approach will allow physicists to produce a big enough amount of simulated data needed by the next HL LHC experiments using limited computing resources.

1 Introduction

Simulation is an important part of studies in particle and nuclear physics from the initial detector design phase to the final comparison with theoretical models. Traditionally, simulations rely on the Monte Carlo methods which require significant computational resources that will not scale to meet the growing demands resulting from large quantities of data expected during High Luminosity Large Hadron Collider (HL LHC) runs. One of the most well known examples of simulation software based on the Monte Carlo methods is `Geant4` [?] implementing cutting-edge heuristics and algorithms. The detailed simulation of particle collisions and interactions using `Geant4` as captured by detectors at the LHC annually requires billions of CPU hours constituting more than half of the LHC experiments' computing resources [? ?]. More specifically, the detailed simulation of particle showers in calorimeters is the most computationally demanding step.

One of the possible approaches to simulate the calorimeter response is using the deep learning techniques. In particular, the Generative Adversarial Networks based approach has been applied sufficiently to the similar task of modeling particle showers in multi-layer Calorimeters with speed-up factors of up to 100 000 [?]. In our case, a particle is described by 5 parameters: 3d momentum (p_x, p_y, p_z) and 2d coordinate (x, y). We propose a new Deep Learning based framework for high-fidelity fast simulation of particle showers in the

---
*e-mail: sayankotor1@gmail.com
**e-mail: egorlova68@gmail.com
***e-mail: eo.zakharov@gmail.com

specific LHCb calorimeter in order to replace the existing Monte Carlo based methods and achieve a significant speed-up factor.

2 Related work

Generative models are of great interest in deep learning. With these models one can approximate a very complex distribution defined by a set of its samples. For example, such models can be utilized to generate a face image of a never existed person or to continue a video sequence given several initial frames. In this section we give a brief overview of the most popular generative model in computer vision — Generative Adversarial Networks (GANs), its strong and weak sides and different modifications to alleviate its weaknesses. In **??** we review analyze an article about applying GANs to the simulation problem in physics.???

2.1 Generative Adversarial Networks

GANs were originally presented by I. Goodfellow *et al*let@tokeneonedotin 2014 [**?**] and quickly became a state-of-the-art technique in areas such as image generation [**?**] with huge number of extensions [**? ? ? ?**].

GANs aim to learn a *generator G* to warp an easy-to-draw distribution $p(z)$ (e.g. $p(z) = \mathcal{N}(0, I)$) into a target distribution $p_{\text{data}}(x)$. In this case the sampling from target distribution $p_{\text{data}}(x)$ is done by first drawing a sample from the distribution $p(z)$ and then feeding it into the generator: $G(z) \sim p_{\text{data}}(x), z \sim p(z)$. The generator is learned by using a feedback from an external classifier (usually called *discriminator*), which tries to find discrepancy between the target distribution $p_{\text{data}}(x)$ and fake distribution defined by samples from the generator $G(z), z \sim p(z)$. The process in summarized in fig. 1.

More formally, generator G and discriminator D are o ... game :

$$\min_G \max_D \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))], \tag{1}$$

where $\mathbf{x} \sim p_{\text{data}}$ are samples from the input data, $\mathbf{z} \sim p(\mathbf{z})$ are the random noise samples, $G(\mathbf{z})$ are the generated images and $D(G(\mathbf{z}))$ is the output of the discriminator specifying the probability of an input being real.

In practice this ojective is optimized using alternating gradient descent. For a fixed generator, the discriminator minimizes binary cross-entropy in a binary classification problem (samples from target distribution versus samples from generator). For a fixed discriminator, generator is updated to make its samples to be misclassified by discriminator.

2.1.1 Wasserstein GAN

There are various modifications of GANs for struggling with some typical problems and for improving the training procedure. A common GANs issue is so–called mode collapse when $p_{\text{model}}(\mathbf{x})$ fails to capture a multimodal nature of $p_{\text{data}}(\mathbf{x})$ and in extreme cases all the generated samples might be identical, in more involved architectures such as Waserstain GAN [**?**] the discriminator loss is argued to be consistent with the image quality. The main idea is to apply the Wasserstein-1 distance in order to compare p_{data} and p_{model}.

Let $\mathbb{X}$ be a compact metric set and let $\mathbb{B}$ denote the set of all the Borel subsets of $\mathbb{X}$. Let *Prob*($\mathbb{X}$) denote the space of probability measures defined on $\mathbb{X}$. The *Earth-Mover* or *Wasserstein-1* distance between two distributions $\mathbb{P}_r, \mathbb{P}_g \in Prob(\mathbb{X})$ is defined in the following way:
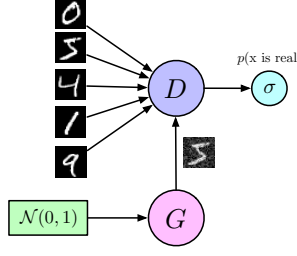
Figure 1: Generative Adversarial Networks for digit generation. The generator G transforms the noise vector $\mathbf{z} \sim p(z)$ to an image of a digit and the discriminator D classifies inputs as real digits or fake digits from generator. Generator and discriminator are trained in an adversarial manner: the task of G is to make it imporssible for D to distinguish between the real and fake digits as in this case G reproduces the data distribution p_{data}.

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y) \sim \gamma}[\|\mathbf{x} - \mathbf{y}\|], \tag{2}$$

where $\Pi(\mathbb{P}_r, \mathbb{P}_g)$ is the set of all joint distributions $\gamma(x, y)$ whose marginals are respectively $\mathbb{P}_r$ and $\mathbb{P}_g$. $\gamma(x, y)$ denotes how much "mass" must be transported from x to y in order to transform the distributions $\mathbb{P}_r$ into the distribution $\mathbb{P}_g$. Thus, the Wasserstein distance can be defined as the minimum cost of transporting mass in order to transform the distribution $\mathbb{P}_r$ into the distribution $\mathbb{P}_q$.

By applying to this distance the *Kullback–Leibler divergence*, the *Jensen-Shannon divergence* and the Kantorovich–Rubinstein duality and also parameterization of family of functions $\{f_w\}_{w \in \mathcal{W}}$ that are all K-Lipschitz for some K we could consider solving the problem (for more detailed information see the paper [**?**])

$$\max_{w \in \mathcal{W}} \mathbb{E}_{x \sim P_r}[f_w(x)] - \mathbb{E}_{z \sim p(z)}[f_w(g_\theta(z))] \tag{3}$$

and if the supremum in (3) is attained for some $w \in \mathcal{W}$, this process would lead to a calculation of $W(\mathbb{P}_r, \mathbb{P}_\theta)$ up to a multiplicative constant. So, the WGAN value function is

$$\min_G \max_{D \in \mathcal{D}} \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_r}[D(\mathbf{x})] - \mathbb{E}_{\tilde{\mathbf{x}} \sim \mathbb{P}_g}[D(\tilde{\mathbf{x}})], \tag{4}$$

where $\mathcal{D}$ is the set of 1-Lipschitz functions and $\mathbb{P}_g$ is the model distribution defined by $\tilde{\mathbf{x}} = G(\mathbf{z})$, $\mathbf{z} \sim p(\mathbf{z})$.

In general, the training procedure of GANs is known to be difficult and presents such issues as mode collapse, and WGAN often helps to overcome this problem.

The Wasserstein GAN value function makes optimization process easier because the discriminator's gradient with respect to its input is better behaved than its GAN counterpart. Also, it was noticed that the WGAN value function tend to correlate with the original data quality which is not always the case for ordinary GANs.

2.1.2 Wasserstain GAN with gradient penalty

An alternative way to enforce the Lipschitz constraint was introduced in [**?**]. The authors consider directly constraining the gradient norm of the discriminator's output with respect to its input because a differentiable function is 1-Lipschtiz if and only if it has gradients with

norm at most 1 everywhere. To come over tractability issues a soft version of the constraint with a penalty on the gradient norm for random samples $\tilde{\mathbf{x}} \sim \mathbb{P}_{\tilde{\mathbf{x}}}$ was suggested. Therefore, a new obtained objective is

$$\mathcal{L}(\boldsymbol{\theta}) = \underbrace{\underset{\tilde{\mathbf{x}} \sim \mathbb{P}_g}{\mathbb{E}} \Big[D(\tilde{\mathbf{x}}) \Big] - \underset{\mathbf{x} \sim \mathbb{P}_r}{\mathbb{E}} \Big[D(\mathbf{x}) \Big]}_{\text{original loss}} + \underbrace{\lambda \underset{\mathbf{x} \sim \mathbb{P}_g}{\mathbb{E}} \Big[(\|\nabla_{\tilde{\mathbf{x}}} D(\tilde{\mathbf{x}})\|_2 - 1)^2 \Big]}_{\text{gradient penalty}}. \tag{5}$$

This approach demonstrates strong modeling performance and stability across a variety of architectures. Now it is a state-of-the-art technique in GANs. How to calculate this loss in practice is described on section 3.3.

2.2 GANs in high energy physics

The first systematic study on the application of Deep Learning to Particle Physics was carried out by Paganini et al. in 2017. In the paper [?] the authors investigated application of GANs for detailed particle simulation in the 3-layer unheterogeneous calorimeter at the LHC. `Geant4` was utilized to create the dataset. Positrons, photons and charged pions with various energies are incident perpendicular on the center of the calorimeter front. Energies in the training are uniform in the range between 1 GeV and 100 GeV. The first layer was represented as a 3×96 image, the middle layer as a 12×12 image, and the last layer as a 12×6 image.

The implemented model is called CaloGAN and the main architecture feature is using the locally-connected layers [?] to modify DCGAN structure [?]. The idea of locally connected layers is based on the fact that every pixel position gets its own filter while an ordinary convolutional layer has a set of same parameters for every position. Spreading of this method to particle physiscs simulation was described in the previous work of the authors and such type of neural network was called LAGAN [?]. A special section in the paper is devoted to the evaluation of the quality of the CaloGAN produced images where sparsity level, energy per layer or total energy are suggested to measure the performance of the model.

The obtained results demonstrate a prospect of application of GANs for the particle showers generation and replacing of the Monte Carlo methods with the proposed approach. The CaloGAN shows sizable simulation-time speed ups compared to `Geant4`.

In fact, the CaloGan model is based on DCGAN with the described tricks. However, GANs tend to suffer from mode collapse. Therefore, the CaloGan architecture cannot be applied for all datasets, because there is a high probability of mode collapse appearance and it is a limitation of this work.

3 Our model

Our idea is to treat simulations as a black-box and replace the traditional Monte Carlo simulation with a method based on Generative Adversarial Networks. WGANs with gradient penalty are considered to be state-of-the-art technique for image producing, so we decide to implement a tool based on this particular approach. For it to be useful in realistic physics applications such a system needs to be able to accept requests for the generation of showers originating from an incoming particle parameters such as 3d momentum and 2d coordinate. We introduce an auxiliary task of energy reconstruction to condition on these parameters p_x, p_y, p_z and x, y.

3.1 Dataset

At the work we focused on electrons interactions inside the electromagnetic calorimeter at the LHCb. In particular, the calorimeter used in this study employs "shashlik" technology of alternating scintillating tiles and lead plates. It consists of 5×5 blocks of size 12 cm $\times$ 12 cm, the cell granularity corresponds to each block is 5×5 of size 2 cm $\times$ 2 cm. There are 66 layers in ECAL – 2 mm absorber and 4 mm scintillator. In fact, the shower appears in 3d, but we summarized allocated energies in each layer per cell. This procedure does not obstruct physics analysis and does not inhibit the shower shape. Thus, this information can be represented as 30×30 images Y with the corresponding parameters $(p_x,\ p_y,\ p_z,\ x,\ y)$. Such image example is presented in fig. 3.

The training data set is created as follows. `Geant4` is utilized to generate particles and simulate their interaction with the calorimeter using the `Ftfp_Bert` physics library based on the `Fritiof` and `Bertini` intra-nuclear cascade models with the standard electromagnetic physics package. So information about every event includes the parameters and 30×30 matrix of energies deposited in scintillator for every cell tower Y. Size of the training dataset is 50 000 events, and we have 10 000 events at the test dataset.

3.2 Model architecture

We need to generate a specific calorimeter response to a particle with some parameters. It means that a model is required to be conditional. Firstly, we describe a generator and discriminator architecture. The generator maps from an input (a 512×1 vector sampled from the Gaussian distribution and the particle parameters) to an 30×30 image $\hat{y}$ using deconvolutional layers (in fact, it is an upsampling procedure and convolutions) which are arranged as follows. We concatenate the noise vector and the parameters $(p_x,\ p_y,\ p_z,\ x,\ y)$, after that we add a fully connected layer with reshaping and obtain $256 \times 4 \times 4$ output. After a sequence of 2d deconvolutions we get outputs of size $128 \times 8 \times 8$, $64 \times 15 \times 16$ and $32 \times 32 \times 32$ with ReLu activation functions. After this procedure we crop the last output to obtain the image of desired size 30×30.

As for the discriminator, it takes a batch of images as input (all images in the batch are real or generated by G) and returns the score $D(\mathbf{y})$ or $D(\hat{\mathbf{y}})$ as it is described in [**?**]. Discriminator architecture is simply the reversed generator architecture (i.e. sizes of layers go in the opposite order). It implies that we have 30×30 matrix as input, then we obtain layers outputs of size $32 \times 32 \times 32$, $64 \times 15 \times 16$, $128 \times 8 \times 8$, after that reshaping leads to $256 \times 4 \times 4$, and by applying LeakyRelu activation function we get the final score. The model scheme is presented in fig. 2.

How to train WGAN with gradient penalty in conditional manner is described in the following section.

3.3 Training strategy

Due to the nature of WGAN loss, conditioning on the continuous value is a non-trivial task. To overcome this issue we suggest to embed a pretrained regressor in our model. We train a neural network to predict the particle parameters by the calorimeter response. As for architecture, it has the same one as the discriminator but with a perceptual loss described in [**?**] because, as we figured out, it works better rather than standard MSE. By building up the information from the pretrained regressor into the discriminator gradient, we obtain the conditional model because we train the generator and the discriminator together. Now the discriminator makes the generator produce a specific calorimeter response.

$$\min_{G} \ \mathbb{E}_{\hat{x}\sim\mathbb{P}_{\text{fake}}}[-D(\hat{x})] + ||y - \hat{y}||_1$$

y

input
5x1:
px, py, pz, ...

concat

noise
Nx1

Generator

256x4x4
128x8x8
64x16x16
32x32x32 30x30

FC + reshape

— Upsampling 2x + Conv + BN + ReLU

CxHxW — output tensor size (w/o batch size)

CxHxW CxHxW — Conv s2 + LeakyReLU (gray = fixed)

x real

$\hat{x}$ fake

Discriminator

30x30 32x32x32
64x16x16
128x8x8
256x4x4

$D(x)$
score
1x1
$D(\hat{x})$

Regressor (pretrained)

30x30 32x32x32
64x16x16
128x8x8
256x4x4

$\hat{y}$
input
5x1

$$\min_{D} \ \mathbb{E}_{\hat{x}\sim\mathbb{P}_{\text{fake}}}[D(\hat{x})] - \mathbb{E}_{x\sim\mathbb{P}_{\text{real}}}[D(x)] \\ + \lambda\mathbb{E}_{\tilde{x}\sim\mathbb{P}_{\tilde{x}}}[(||\nabla_{\tilde{x}}D(\tilde{x})||_2 - 1)^2]$$
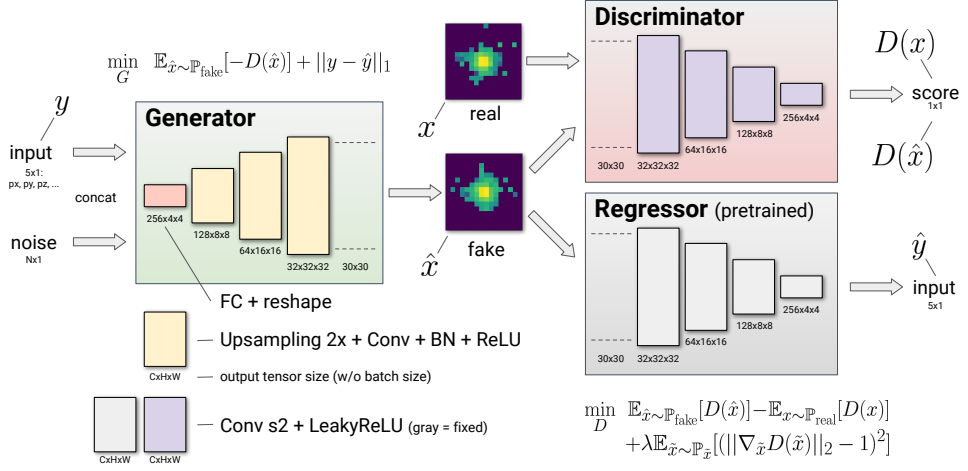
Figure 2: Model architecture. Pretrained regressor for the particle parameters prediction makes our model conditional. Thanks to building up the information from the pretrained regressor into the discriminator gradient we learn G to produce a specific calorimeter response.

Matrices from our dataset are pretty sparse because almost all information is located in central cells (see fig. 3). To make optimization process easier we apply a box–cox transformation. This mapping helps to smooth the data that makes the optimization process more stable. Results obtained with the described model are presented in the following section.

4 Results

From a series of simulated electron showers our conditional GAN is tasked with learning the simulated data distributions generated by Geant4. There exist several methods to estimate the performance of generative models, but not all evaluation criteria are equally suitable and reliable for all applications. In our paper we focus on quantitative evaluation based on physics-driven similarity metrics. The choice reflects the domain specific procedure for data-simulation comparison.

Examples of real and produced images are presented in fig. 4b. It can be concluded that produced images look quite similar to the original data but we are also supposed to do more deep analysis.

As a part of the analyzing generated images quality we calculated some physical characteristics of shower shapes such as cluster asymmetry, shower width and sparsity level. The obtained characteristics are presented in fig. 5. How to calculate these variables is described in Appendix.

The results are in good agreement with fully simulated data. Energy showers are faithfully reproduced and show a reasonable agreement with the standard Monte Carlo techniques. A detailed study to asses the quality of the GAN images using typical high level calorimeter reconstruction variables confirms this statement.

As for model performance, we trained our model for 3000 epochs which takes about 70 hours on GPU NVIDIA Tesla K80. The sampling rate is 0.07 ms per sample on GPU, 4.9 ms per sample on CPU.
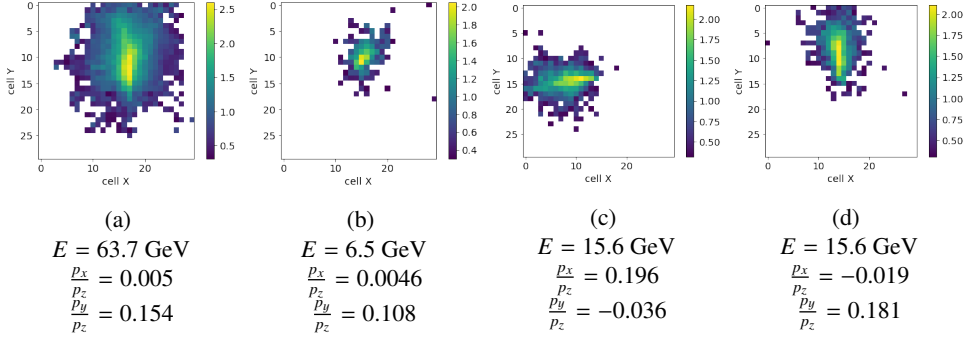
(a)

$E = 63.7$ GeV

$\frac{p_x}{p_z} = 0.005$

$\frac{p_y}{p_z} = 0.154$

(b)

$E = 6.5$ GeV

$\frac{p_x}{p_z} = 0.0046$

$\frac{p_y}{p_z} = 0.108$

(c)

$E = 15.6$ GeV

$\frac{p_x}{p_z} = 0.196$

$\frac{p_y}{p_z} = -0.036$

(d)

$E = 15.6$ GeV

$\frac{p_x}{p_z} = -0.019$

$\frac{p_y}{p_z} = 0.181$

Figure 3: Images and the parameters from the real dataset. TODO: generate high quality images

(a)

$E = 63.7$ GeV

$\frac{p_x}{p_z} = 0.005$

$\frac{p_y}{p_z} = 0.154$

(b)

$E = 6.5$ GeV

$\frac{p_x}{p_z} = 0.0046$

$\frac{p_y}{p_z} = 0.108$

(c)

$E = 15.6$ GeV

$\frac{p_x}{p_z} = 0.196$

$\frac{p_y}{p_z} = -0.036$

(d)

$E = 15.6$ GeV

$\frac{p_x}{p_z} = -0.019$

$\frac{p_y}{p_z} = 0.181$
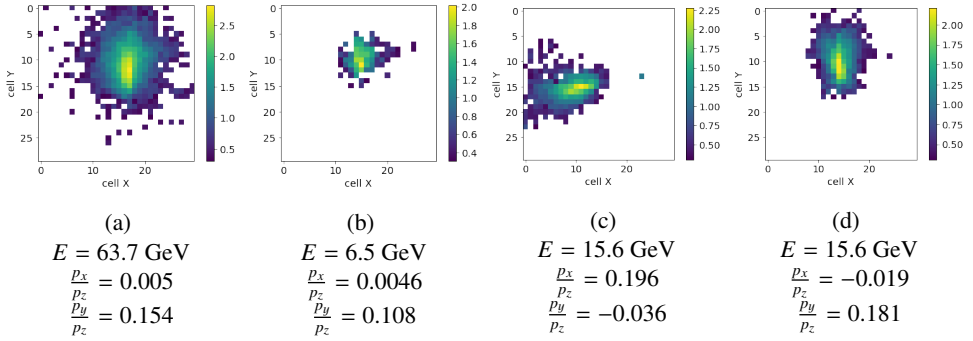
Figure 4: Images and the parameters from the real dataset. TODO: generate high quality images

(a) The transverse asymmetry of real and generated images

(b) The longitudal asymmetry of real and generated images

(c) The transverse width of real and generated images

(d) Delta X
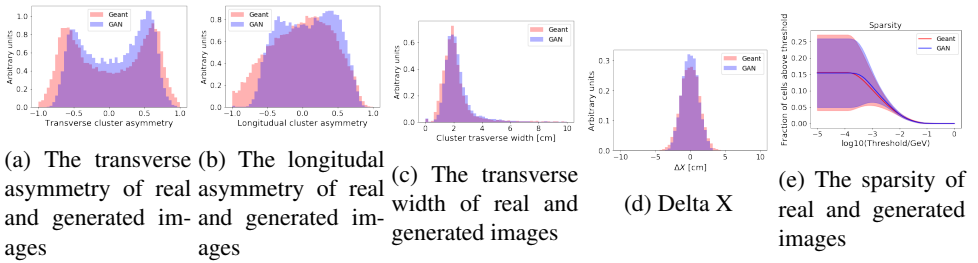
(e) The sparsity of real and generated images

Figure 5: Generated images quality evaluation including described physical characteristics.

5 Conclusion and future outlook

The research proves that Generative Adversarial Networks are a good candidate for fast simulating of high granularity detectors typically studied for the next generation accelerators. We have successfully generated images of shower energy deposition with a condition on the particle parameters such as the momentum and the coordinate using modern generative deep neural network techniques such as Wasserstain GAN with gradient penalty.

Future work will be focused improving obtained results such as shower shape characteristics.

References

[1] S. Agostinelli, J. Allison, K.a. Amako, J. Apostolakis, H. Araujo, P. Arce, M. Asai, D. Axen, S. Banerjee, G.. Barrand et al., Nuclear instruments and methods in physics research section A: Accelerators, Spectrometers, Detectors and Associated Equipment **506**, 250 (2003)

[2] C. Bozzi, Tech. rep. (2014)

[3] J. Flynn, Tech. rep. (2015)

[4] M. Paganini, L. de Oliveira, B. Nachman, arXiv preprint arXiv:1705.02355 (2017)

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, *Generative adversarial nets*, in *Advances in neural information processing systems* (2014), pp. 2672–2680

[6] A. Radford, L. Metz, S. Chintala, arXiv preprint arXiv:1511.06434 (2015)

[7] M. Arjovsky, S. Chintala, L. Bottou, arXiv preprint arXiv:1701.07875 (2017)

[8] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A.C. Courville, *Improved training of wasserstein gans*, in *Advances in Neural Information Processing Systems* (2017), pp. 5769–5779

[9] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, *Deepface: Closing the gap to human-level performance in face verification*, in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2014), pp. 1701–1708

[10] L. de Oliveira, M. Paganini, B. Nachman, arXiv preprint arXiv:1701.05927 (2017)

[11] J. Johnson, A. Alahi, L. Fei-Fei, *Perceptual losses for real-time style transfer and super-resolution*, in *European Conference on Computer Vision* (Springer, 2016), pp. 694–711